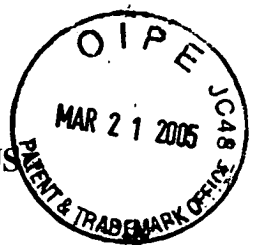


DOCKET NO. 1236 - US

REFERENCE AC INFORMATION DISCLOSURE STATEMENT OF:

Applicant(s): Chi Leung Lau et al

Serial No.: 09/909,645

Filed: 07/20/2001

Title: IP Flow Discovery for IP Probe Auto-Configuration and SLA Monitoring

THIS PAGE BLANK (USPTO)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 May 2001 (25.05.2001)

PCT

(10) International Publication Number
WO 01/37493 A1

(51) International Patent Classification[7]: H04L 12/56

(21) International Application Number: PCT/EP00/10693

(22) International Filing Date: 31 October 2000 (31.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
99122900.6 18 November 1999 (18.11.1999) EP

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventors: BAUCKE, Stephan; Ericsson Allee 1, 52134 Herzogenrath (DE). LUDWIG, Reiner; Maubacher Strasse 4, 52393 Hürtgenwald (DE). MEYER, Michael; Annastrasse 17, 52062 Aachen (DE).

(74) Agent: MOHSLER, Gabriele; Ericsson Eurolab Deutschland GmbH, Ericsson Allee 1, 52134 Herzogenrath (DE).

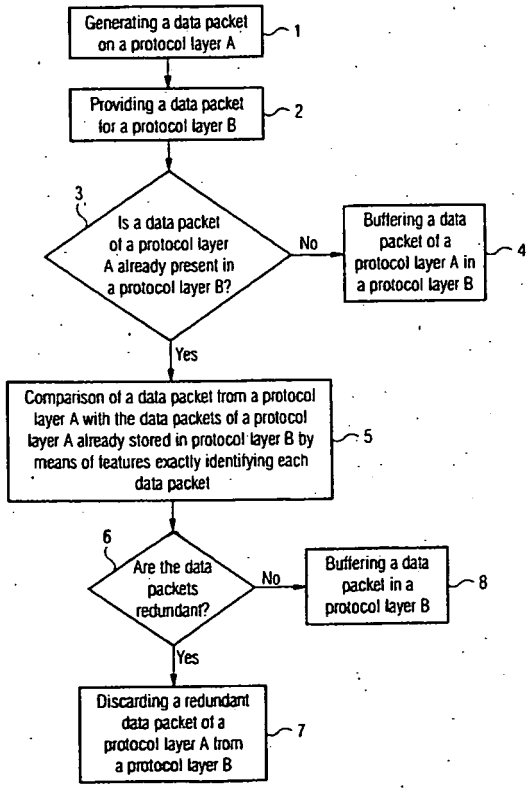
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

[Continued on next page]

(54) Title: METHOD AND DEVICE FOR IMPROVING A DATA THROUGHPUT IN A COMMUNICATION SYSTEM



(57) Abstract: The invention relates to a method and a device for increasing a data throughput in a data transmission via a communication network. In this respect the data packets are examined for redundant data packets on the link layer (5). In case of a temporary transmission failure, the transmission of unacknowledged data packets is repeated on the transport layer (1). Said data packets are provided to the link layer (2). It is the task of the link layer to examine whether the released data packet is a redundant data packet of a data packet already present on the link layer (5). If the redundancy of the data packets is found, the redundant data packet is discarded (7). The data packet is stored on the link layer, if either no data packets are already present on the link layer (4) or if the data packets are not redundant (8).

WO 01/37493 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Method and device for improving a data throughput in a communication system

The invention relates to a method and a device for increasing a data throughput in a data transmission via a communication network.

Especially in communication networks having a low transmission rate it proves to be necessary to develop methods efficiently utilizing the low transmission rates between a sender and a receiver. Communication networks of this type include mobile cellular networks such as GSM (Global System for Mobile Communication) or GPRS (General Packet Radio Service).

The communication between partner instances, for example between a sender and a receiver, is realized by means of a protocol stack. A protocol stack comprises several protocol layers lying on top of each other, whereby a communication between the neighboring protocol layers takes place. The data received or made available for the transmission are released from one layer to the directly neighboring layer. It is the task of a layer, among others, to process the data, whereby the segmentation thereof is one part of the processing. Frequently the data size of a layer exceed the size of the data packets which can be transmitted via a physical connection. For this reason the data is divided into smaller data packets which are successively arranged for a transmission. In addition, the data are formatted on each protocol layer. The formatting of the data includes particularly the addition of the control data characteristic for each protocol layer. Usually the control data are attached at the beginning of a data packet in the form of the so-called header and/or at the end in the form of the so-called tail. The actual data are contained in the user field.

In the following the processing of the data packet is explained in more detail by means of the TCP/IP protocol stack.

The TCP/IP (Transmission Control Protocol/Internet Protocol) protocol stack is the standardized protocol stack for internet applications. Said protocol stack comprises five protocol layers. The uppermost layer, the application layer, comprises the application directly used by a user, e.g. WWW, FTP or email. The application layer communicates with transport layer arranged thereunder. The tasks of said layer are fulfilled by transport protocols such as TCP (Transmission Control Protocol) or UDP (User Datagram Protocol). In the following only TCP will be dealt with more closely, as this transport protocol is used for the communication of non-real time applications such as WWW, FTP or email. The network layer including the internet protocol (IP) is arranged underneath the transport layer. The two lowermost layers, the link layer and the physical layer, can be combined under the term of network-oriented layer, since they are specifically defined in response to the underlying network.

The transport protocol TCP offers a reliable transport service of data between two communicating partners. Reliability in this respect refers to the avoidance of errors, maintenance of sequence and protection against data losses and duplicates. The transport protocol TCP was originally drafted for the application in fixed networks. In order to meet the conditions of a fixed network, a number of laborious algorithms were implemented in TCP, for example, for avoiding the occurrence of congestions in network nodes or overload situations of the network. The individual mechanisms such as the so-called sliding window mechanism, the cumulative acknowledgment scheme, the so-called Slow-Start-Algorithm or the so-called Congestion Avoidance Algorithm, will not be described in more detail. The exact description of said methods can be inferred from „TCP/IP. Illustrated, Volume 1" by W. Richard Stevens.

A method, which is hereinafter entered into more closely, refers to the detection and correction of packet losses.

The principal mechanism of TCP detecting the loss of data packets, consists in the use of time alarms, the so-called timeouts. A timeout is connected with a time interval provided for the implementation of a certain task. If the task is not realized within said time interval, a timeout occurs. In the TCP, for example, a time interval - the so-called RTT (Round Trip Time) - is defined, which specifies the time elapsing between the transport of a TCP packet to the IP layer and the receipt of a pertinent acknowledgment message from a receiver that said data packet was correctly received. Said time interval allows the adjustment to different transmission conditions. An RTO (Retransmission Timeout) is calculated on the basis of the average value and the variance of the RTT. An RTO relates to the time at which a retransmission of data packets takes place, for which no acknowledgment message on the receipt of said data packets has been received. The exact calculation of the RTO can be inferred from RFC 793, Transmission Control Protocol, September 1981. In view of the timeout-based error recognition the accuracy of the calculation of the RTO largely influences the throughput. If the timeout has been assessed too short, unnecessary data packets will be retransmitted. If, on the other hand, it is too long, too much time passes until the loss of a data packet is noticed and the data packet can be retransmitted. The calculation of an RTT is done dynamically during a transmission in order to capture the deviations in a transmission. Particularly problematical are networks having a specially high error rate in a data transmission thereby being characterized by high deviations in the transmission rates. Mobile cellular networks, such as GSM (Global System for Mobile Communication), GPRS (General Packet Radio Service), UMTS (Universal Mobile Telecommunication System), EDGE (Enhanced Data rates for GSM Evolution) or IMT-2000, are part of these networks.

An error recognized in the transmission of a data packet is corrected by a method for securing a reliable service. One example for said method is the ARQ (Automatic Repeat Request). By this method a copy of the data packet is prepared and stored on the sending side for each sent data packet, until the receiver acknowledges the correct receipt of the sent data packet. The receiver checks the received data packet and acknowledges the correct receipt by sending an acknowledgment message, or he discards the data packet and informs the sender about the necessity of retransmitting the same data packet. Said information either takes place by sending a message, or when the corresponding time interval for the receipt of a positive message at the sending side has expired, e.g. the RTO. In this case the sender takes the copy of the data packet from the buffer and transmits it anew. Thus, the previous errors are corrected.

The strategy of the exponential enlargement of the RTO, the so-called exponential backoff, used by the TCP results in a deterioration of the transmission throughput. Said strategy is used for dealing with overload situations. Due to the fact that in general transmission errors rarely occur on the wire-bound transmission links, for which TCP was originally designed, the major part of all packet losses is due to congestions in network nodes caused by overload. For this reason strategies for avoiding the occurrence of congestions such as the so-called exponential backoff, are used in TCP. By means of this method the RTO is set at double the value of the previous RTO value once an error occurs. Assuming that the first retransmission of a non-received data packet takes place 1.5 seconds after the data packet was initially sent. Thereafter, the RTO is increased by double the value, i.e. a retransmission takes place after 3, 6, 12, 24, 48 and 64 seconds. The value then remains constant, i.e. always after 64 seconds a new attempt will be made to retransmit the unacknowledged data packet. 64 seconds is the upper limit specified in the implementation of TCP/IP in the operating system Unix 4.3BSD by

the Computer Systems Research Group, University of California, Berkley.

The above-described example shows that the RTO rapidly grows if several successive retransmissions get lost during a temporary failure on the transmission route, whereby in the most unfavorable case a retransmission of a data packet takes place after 64 seconds. In other words, a long time may elapse until the retransmission takes place once the temporary failure has been removed.

For this reason it is more efficient to scan an unavailable connection on the link layer. The link layer is responsible for the so-called point-to-point connection, i.e. for the connection between two directly communicating network nodes. Therefore, it is not necessary on the link layer to provide mechanisms for avoiding overload situations in network nodes, as is the case in TCP. For this reason a TCP packet, which was previously formatted to an IP packet on the network layer and then released to the link layer, is immediately made available for the transmission on the link layer when the transmission link is working again. The time during which a data packet is intermediately stored on the link layer depends on the underlying communication network. If the time, however, is correspondingly long, so that a data packet is retransmitted on the TCP layer, several duplicates of a data packet are buffered on the link layer in case of a lasting malfunction.

Assuming the connection is not working for several minutes. This can happen if, for example, a subscriber is in an environment in which a radio connection is impossible, e.g. in a garage. In this case TCP attempts to retransmit a data packet for 9 minutes. If the above-described example mentioning the times at which a retransmission takes place is used in this case, it means that 12 retransmissions of the same data packet take place for 9 minutes. Accordingly 12 copies of the same data packet are buffered on the link layer. This happens on the

basis of the assumption that the time, during which the data packets are stored on the link layer, is correspondingly long.

Accordingly it is an object of the present invention to provide a method and device that guarantees a more efficient provision of data at the sending side for an improved throughput in the data transmission in connection with packet-oriented applications.

According to the invention this object is provided by the teaching of patent claims 1 and 17 and by the teaching of patent claim 20.

It has thereby proved to be an advantage that the data throughput in the network is increased, for, due to the reduction of the number of redundant data packets or even of duplicates, the multiple transmission of an identical data packet via a communication network is not necessary.

Another advantage relates to the reduction of the memory space on the sending side as well as on the receiving side, which is achieved by discarding redundant data packets on the sender's link layer.

Further advantageous embodiments of the invention are disclosed in patent claims 2 to 16 and 18 and 19.

In the following the invention is explained in more detail by means of embodiments and figures, wherein

Fig. 1 shows a flow chart of the inventive method,
Fig. 2 shows an illustration of a GPRS protocol stack in a terminal,
Fig. 3 shows an illustration of a TCP header,
Fig. 4 shows an illustration of an IP header.

In the following the invention is explained in more detail by means of figure 1 and patent claim 1.

In a first device, for example a sender, data packets are generated from application data on the basis of a hierarchic structure of a protocol stack, in order to provide the application data for a transmission. According to figure 1 the data packets of a first protocol layer are generated in a first step 1, which layer will hereinafter be designated as protocol layer A. Said data packets are exactly identified by means of corresponding features. Upon the generation of the data packets, they are provided for at least another protocol layer hereinafter designated as protocol layer B, 2. If no data packets of protocol layer A already exist in protocol layer B, the data packet released from protocol layer A is stored, 4. If it is, however, found out in protocol layer B, 3, that in said protocol layer data packets of protocol layer A are already provided, it is compared in step 5 by means of the features exactly identifying each data packet whether the data packet present in protocol layer B and the just released data packet are redundant. The number of comparisons to be made thereby depends on the number of data packets of protocol layer A present in protocol layer B. If a redundancy of two data packets is found, 6, a data packet is discarded, 7. The data packet of protocol layer A is stored in protocol layer B, if the data packets are not redundant, 8. In this respect data packets are called redundant, if they are either identical or if it is found on the basis of the features exactly identifying each data packet that one data packet is contained in another one. In this case the data packet contained in the other one is discarded. Moreover, data packets of protocol layer A, when released to protocol layer B, can be sub-divided into smaller data packets of protocol layer B. In this case data packets are found to be redundant through the comparison of the data packet of protocol layer A with the data packets of protocol layer B, which in combination result in a data packet of protocol layer A. If a data packet just released from protocol layer A and a

data packet of protocol layer A sub-divided into smaller data packets on protocol layer B are found to be redundant, either the just released data packet of protocol layer A is discarded, or the data packets of protocol layer B forming the redundant data packet of protocol layer A are discarded.

In the following the inventive device according to claim 17 is explained in more detail (without figure). In the inventive device, data packets of protocol layer A are generated on a first protocol layer, protocol layer A, with means for generating data packets. The means for providing data packets releases the data packets of protocol layer A for at least an additional protocol layer, protocol layer B. Said data packets are stored on protocol layer B with means for buffering the data packets, unless a redundant data packet already exists on protocol layer B. The redundancy of data packets is found with means for comparing data packets. The comparison is carried out by means of the features exactly identifying each data packet. The transport of the data packets provided for the transmission is realized with sending means. The receipt of a data packet by a receiver is acknowledged by sending an acknowledgment message. The receipt of said message at the sending side is realized with receiving means.

The present invention is used in each architecture, in which in a sending unit at least two separately operating methods for providing a reliable service are integrated in a protocol stack. The task of a method for providing a reliable service is, for instance, fulfilled by the ARQ method. In the ARQ method a copy of each transmitted data packet is generated and stored at the sending side. The copy of the sent data packet is deleted from the buffer, if an acknowledgment message on the correct receipt of the message is received from a receiver. If no acknowledgment message is received or if a negative acknowledgment message is received, the same data packet is retransmitted. The missing acknowledgment receipt and the retransmission of a data packet are controlled by means of so-

called timeouts. The invention is applied in an architecture, in which methods for providing a reliable service are integrated in a protocol stack on different protocol layers forming part of different protocols, for example in a protocol stack, where the TCP is available on the transport layer and the RLC (Radio Link Control) on the link layer, or where a TCP is introduced over another TCP. The latter case applies to a GGSN node in a GPRS network. A more detailed description can be inferred from GSM 03.60 Version 6.3.0.

In the following the invention is explained in more detail by means of an embodiment based on the GPRS communication network.

One example relating to a network, in which the invention can be used, is the GPRS (General Packet Radio Service). GPRS is a packet-oriented network the principal task of which consists in an improved support of internet applications. For describing the invention, the GPRS protocol stack implemented in the terminal of a user is hereinafter illustrated by means of figure 2. The left-hand side of figure 2 shows a protocol stack with the corresponding protocols. The protocols on the physical layer, which can be seen on the right-hand side of figure 2, are designated Phy. The protocols of the physical layer will not be dealt with in more detail. MAC, RLD, LLC and SNDCP constitute the protocols of the link layer. The following embodiment is based on said protocols. The internet protocol (IP) represents a protocol of the network layer, and TCP (Transmission Control Protocol) a protocol of the transport layer. The application layer comprises available applications.

In the following the protocol stack illustrated by figure 2 is explained in more detail.

As was already mentioned, the application layer forms the uppermost layer with the applications running on said layer such as WWW (World Wide Web). Said layer directly communicates with the transport layer, in this case the TCP, the task of

which is to provide a reliable service. For this purpose the application data are first sub-divided into TCP data packets. A TCP data packet comprises a header and a data field. The data field comprises part of the application data. The size of said field depends on the selected size of the TCP packet. The structure of a TCP header is shown in figure 3. A TCP message header comprises data among others required for the fragmentation and the error recognition. Figure 3 illustrates a complete TCP header with the control data contained therein. In view of the present invention the fields source port number, destination port number and the sequence number are particularly significant and will hereinafter be described in more detail. The sequence number identifies the position of the data of the data packet in the corresponding data flow. For this purpose an initial value for the sequence number of the respective data flow is declared when a connection is set up. Thereafter the sequence number for each byte is increased by one. The position of a byte in the data flow results from the difference of the sequence number of said byte and the initially declared sequence number. The sequence numbers thereby allow the combination of the received data packets at the receiving side in the correct order. The source and destination port number identify the sending and the receiving application.

Moreover, different timeouts are started for a TCP data packet, for example, the RTO (Retransmission Timeout) is among others set for the data packet on the TCP layer upon the transport of the data packet to the network layer. An additional processing of the data takes place on the network layer.

The tasks of the network layer are fulfilled by the internet protocol, the so-called IP. The IP has the property that the exchange of data is unreliable, i.e. the IP does not guarantee that every IP data packet reaches the receiver. Also IP data packets can be received by the receiver in an incorrect sequence or as duplicates. It is the task of TCP to detect the

faulty transmission and to correct the errors. The errors can be detected in different ways, as was already explained. The correction of the error is done through a retransmission of the faulty data packet. For this purpose the data packets released to the network layer are intermediately stored on the TCP protocol layer. A data packet is only deleted from the buffer at the sending side when it has been secured that the data packet was really correctly received by the receiver, which takes place after the receipt of an acknowledgment message.

An important task of IP lies in the formatting of the TCP data packets, more exactly in the packing of the TCP data packets into IP data packets. The IP data packets generated by IP are called IP datagrams. According to figure 4 an IP datagram consists of a header containing the control data and of a data field in which a TCP data packet is embedded. The feasibility of the invention is based on the examination of an IP data packet and on the comparison with an additional IP data packet. In this respect particularly the fields of an IP data packet guaranteeing an exact identification of each packet are examined. For this reason said fields are referred to as features exactly identifying each data packet, and they will hereinafter be dealt with in more detail.

The structure of an IP header is connected with the used version of the internet protocol. In the following two versions of the IP, the IP version 4, the IPv4, and the IP version 6, the IPv6, will be examined as examples. Figure 4 illustrates an IPv4 datagram.

For the sake of completeness figure 4 shows all fields of the IP header. The fields protocol, source address and destination address are particularly relevant for the invention. The protocol field expresses the protocol type of the data flow in question, for example TCP. When comparing data packets, said fields are examined first. In the case of non-congruence it can directly be concluded therefrom that said data packets cannot

be redundant and that a comparison of further fields is not necessary. The fields source address and destination address correspondingly comprise the addresses of the receiver and the sender. In the case where said fields correspond with each other in view of the data packets to be compared, another examination has to be performed in the TCP header in order to secure the redundancy or even identity of the data packets. This is necessary because in most cases a user uses several applications at the same time during a session, for example, when using electronic mail and surfing in the internet simultaneously. Due to the fact that both applications are possibly carried out on one and the same terminal requesting access to one and the same server, both the destination address and the source address in the data packets to be compared correspond with each other. However, different port numbers are allocated for the different applications and thus for the data flows. Said port numbers identify a corresponding data flow on the transport layer. The information in view of the port number are comprised in the header of a TCP packet, and when differentiating the data flow they are compared with each other in order to exactly determine the differences between the data flows. Only when the port numbers of the destination in view of the data packets to be compared correspond with the port numbers of the source, is the data flow identical. The identity of the data flow is checked by comparing the protocol type, the destination and the source address and the port number of the destination and the source. The redundancy and even the identity of the data packets is determined by means of an additional comparison of the sequence number in the TCP header and the fields having the total length in the IP header. The field total length includes the entire IP datagram, i.e. the header and the data. When dividing the data into packets, each packet receives - as was already described - a precise continuous sequence number. The sequence number corresponds to the first data byte contained in the data field, which data byte indicates the position of the data byte relative to the first byte in the data flow. For this purpose an initial value

for the sequence number is declared when setting up the connection. Thereafter the sequence number for each byte in the data flow is increased by one. The sequence number of a following data packet is determined by means of the sequence number and the total length of the preceding data packet, and the position of a byte in the data flow results from the difference between the sequence number of said byte and the initial sequence number. By means of said data the redundancy of a data packet is determined, i.e. it is determined whether a data packet is contained in another one and whether they are even identical.

The described method for identifying the redundancy of a data packet is implemented in the presently used fourth version of the IP, the IPv4. In the next version of the internet protocol, in the internet protocol version 6, the IPv6, said method is based on the same principle, i.e. on the comparison of the information comprised by the data packet. An allocation of a data packet to a data flow can be realized more efficiently by means of data flow identifiers, so-called flow label. The major aim of the flow labels consists in a faster allocation of a data packet to a data flow, for example, in order to reduce the processing time in the intermediate nodes. Apart from that, a header of an IPv6 datagram equally contains a destination and source address, which can be examined. Moreover, the IPv6 packet contains a TCP data packet, which can be analyzed in the same way as was already described.

Upon the generation of the IP datagrams, the latter are transferred to the directly underlying layer, the link layer. The structure of the link layer depends on the underlying network. According to figure 2 the link layer in GPRS consists of several intermediate layers including the corresponding protocols.

The SNDCP (SubNetwork Dependent Convergence Protocol) provides convergence functionality to map different protocols of the

higher layers on a single link supported by the LLC (Logical Link Control) lying underneath the SNDCP. Said SNDCP includes, among others, the multiplexing of the data packets from the different protocols and the header compression, for example of the TCP/IP header. A segmentation of the data into packets equally takes place on said layer. Usually, however, the size of the SNDCP packets corresponds to the size of the IP packets.

The IP packets are buffered on said layer as long as the connection is not available for a transmission, for example, due to a temporary failure. If the time of the failure is correspondingly long so that the RTO value meanwhile runs on the TCP layer, a new transmission of an unacknowledged TCP data packet is initiated. For this purpose the TCP takes the data packet from its buffer and transfers it to the IP layer which generates an IP packet therefrom. Said packet again is transferred to the SNDCP. The SNDCP examines the IP packet, in particular the fields in the IP and TCP header, and compares said fields with the corresponding fields of the data packets present on said layer. If it is found that an identical IP packet is already present on the SNDCP layer, one of the two packets is discarded. If the IP packet is not a redundant data packet of an already existing data packet, it is provided to the SNDCP layer and kept on said layer until the connection is again ready for a transmission.

According to the hierarchic structure of a protocol stack the data packets are transferred from the SNDCP layer to the directly underlying layer, and according to figure 2 this is the LLC (Logical Link Control) layer. Due to the fact that the underlying network is characterized by high error rates in the transmission, protocols were developed on the link layer, which provide a reliable transmission via a physical connection. The LLC protocol can either operate in a so-called unreliable mode, in which packet losses are not taken into account, or in a so-called reliable mode guaranteeing a secured transport of data packets by means of repeating faulty data packets. The

invention is mainly used in the reliable mode, in which the data packets are buffered for being able to take them out of the buffer and transmit them anew in case a transmission error has occurred. Similar to the SNDCP layer, the present invention can equally be used on said layer, on which the SNDCP packets are intermediately stored. Said layer keeps the SNDCP data packets in the buffer, until it is secured that said data packets have correctly been transmitted and received. If in the meantime a retransmission takes place on the transport layer, said data packet is transferred to the IP layer, to the SNDCP layer and consequently to the LLC layer. By means of features exactly identifying each data packet the received data packet is compared with the data packets present on the LLC layer, and if a redundancy is detected, a redundant data packet is discarded.

The LLC data packets are thereupon transferred to the RLC (Radio Link Control) layer. First, the data packets are sub-divided again into smaller packets, the so-called frames, on said layer. On the basis of these frames the error correction is carried out in the ARQ method in case the RLC operates in the reliable mode. Like with the LLC two modes are supported - the reliable mode, in which the occurred transmission errors are corrected, and the unreliable mode, in which the packet losses are not considered. For the purpose of correcting the errors on the RLC layer, the LLC packets are buffered in the reliable mode. Thus it is also possible that, if a packet is retransmitted, a comparison of LLC packets for finding and discarding redundant LLC packets on the RLC layer is performed.

The embodiment described in the invention relates to a GPRS network. The invention may, however, be used in any other network, in which the data packets undelivered due to the non-availability of a connection are buffered long enough, so that a retransmission of a data packet present on the link layer meanwhile takes place on the higher layers. Due to the fact that the scanning of a meanwhile unavailable connection can

best be performed on the link layer, as in the worst case the scanning on the TCP layer takes place every 64 seconds due to the so-called exponential backoff algorithm, or as the scanning is even impossible in the middle of the transmission route due to the fact that TCP is an end-to-end protocol, it is consequently best to use the invention, particularly to discard the redundant data packets, also on the link layer, because it is guaranteed on said layer that a data packet is directly sent once a connection is again available.

The invention may, however, be used in any other network having a first and at least one additional protocol layer. In this respect it is decisive that the layers are provided with separately operating methods for providing a reliable service, and that the time interval for buffering a data packet on the at least one additional protocol layer is long enough to allow an initiation of a retransmission of a data packet on a first protocol layer resulting in the formation of redundant data packets on the at least one additional protocol layer.

In the embodiment described in the specification, which is based on a GPRS network, the tasks of the first protocol layer are fulfilled by TCP and the tasks of the at least additional one of the protocol layers is fulfilled by the link layer. Both layers are provided with a method for providing a reliable service, which is fulfilled by means of the ARQ method. Other examples for networks, in which the invention can be used, are a GSM (Global System for Mobile Communication), a UMTS (Universal Mobile Telecommunication System), an EDGE (Enhanced Data rates for GSM Evolution) or an IMT-2000 communication network.

1. Method for increasing a data throughput for transmitting data via a communication network between a sender having a first and at least one additional protocol layer and a receiver,

- wherein data packets with exactly identifying features are generated in the first protocol layer (1),
- wherein the data packets are provided for at least an additional protocol layer (2),
- wherein the data packets in the at least one additional protocol layer are compared with already present data packets by means of the features exactly identifying each data packet (5),
- wherein, if a redundant data packet is present in the at least one additional protocol layer, one of the redundant data packets is discarded (7),
- wherein, if a data packet is not present in the at least one additional protocol layer, said data packet is stored (4), (8).

2. Method according to claim 1, wherein the exactly identifying features are the source address, the destination address, the port number of the source, the port number of the destination, the sequence number of a packet and the length of a packet.

3. Method according to claim 1 or 2, wherein the exactly identifying features are comprised in at least one header.

4. Method according to one of the preceding claims, wherein a header is added to a data packet on a protocol layer.

5. Method according to one of the preceding claims, wherein on each protocol layer the data in a data packet are accessed.

6. Method according to claim 1, wherein the receipt of a data packet by a receiver is acknowledged by sending acknowledgment messages.

7. Method according to one of claims 1 to 6, wherein a missing acknowledgment message or the receipt of a negative acknowledgment message results in a retransmission of the same data packet by the sender.

8. Method according to claim 7, wherein the retransmission of the same data packet is made feasible by the preceding generation of a copy of the sent data packet.

9. Method according to one of claims 1, 6, 7 or 8, wherein the copy of a data packet is buffered on a protocol layer until an acknowledgment message on the correct receipt of a data packet is received or a timeout occurs.

10. Method according to one of claims 6 to 9, wherein a retransmission of a data packet is controlled by timeouts.

11. Method according to one of claims 6 to 10, wherein the generation of a copy of a data packet, the receipt of an acknowledgment message and the retransmission of a data packet are secured by means of a method for providing a reliable service.

12. Method according to claim 10, wherein a correction of a faulty transmission of a data packet is performed by means of the method for providing a reliable service.

13. Method according to one of claims 10, 11 or 12, wherein the method for providing a reliable service is an ARQ (Automatic Repeat Request) method.

14. Method according to one of claims 1 or 6 to 13, wherein the time interval for buffering a data packet on the at

least one additional protocol layer is long enough to initiate a retransmission of a data packet on a first protocol layer resulting in the formation of redundant data packets on the at least one additional protocol layer.

15. Method according to one of the preceding claims, wherein a first and at least one additional protocol layer are protocol layers in a protocol stack.

16. Method according to claim 1, wherein the communication network is a GSM (Global System for Mobile Communication), a GPRS (General Packet Radio Service), a UMTS (Universal Mobile Telecommunication System), an EDGE (Enhanced Data rates for GSM Evolution) or an IMT-2000 communication network.

17. Device for increasing a data throughput for transmitting data via a communication network between a sender having a first and at least one additional protocol layer and a receiver, comprising
- means for generating data packets of a first protocol layer,
- means for providing the data packets for at least one additional protocol layer,
- means for comparing data packets,
- means for buffering the data packets of the first protocol layer on the at least one additional protocol layer,
- means for discarding redundant data packets.

18. Device according to claim 17 comprising sending means for sending a data packet via a communication network.

19. Device according to claim 18 comprising receiving means for receiving acknowledgment messages.

20. Use of the method according to claim 1, for increasing a data throughput of a protocol stack providing at least two separately operating methods for providing a reliable service.

FIG. 1

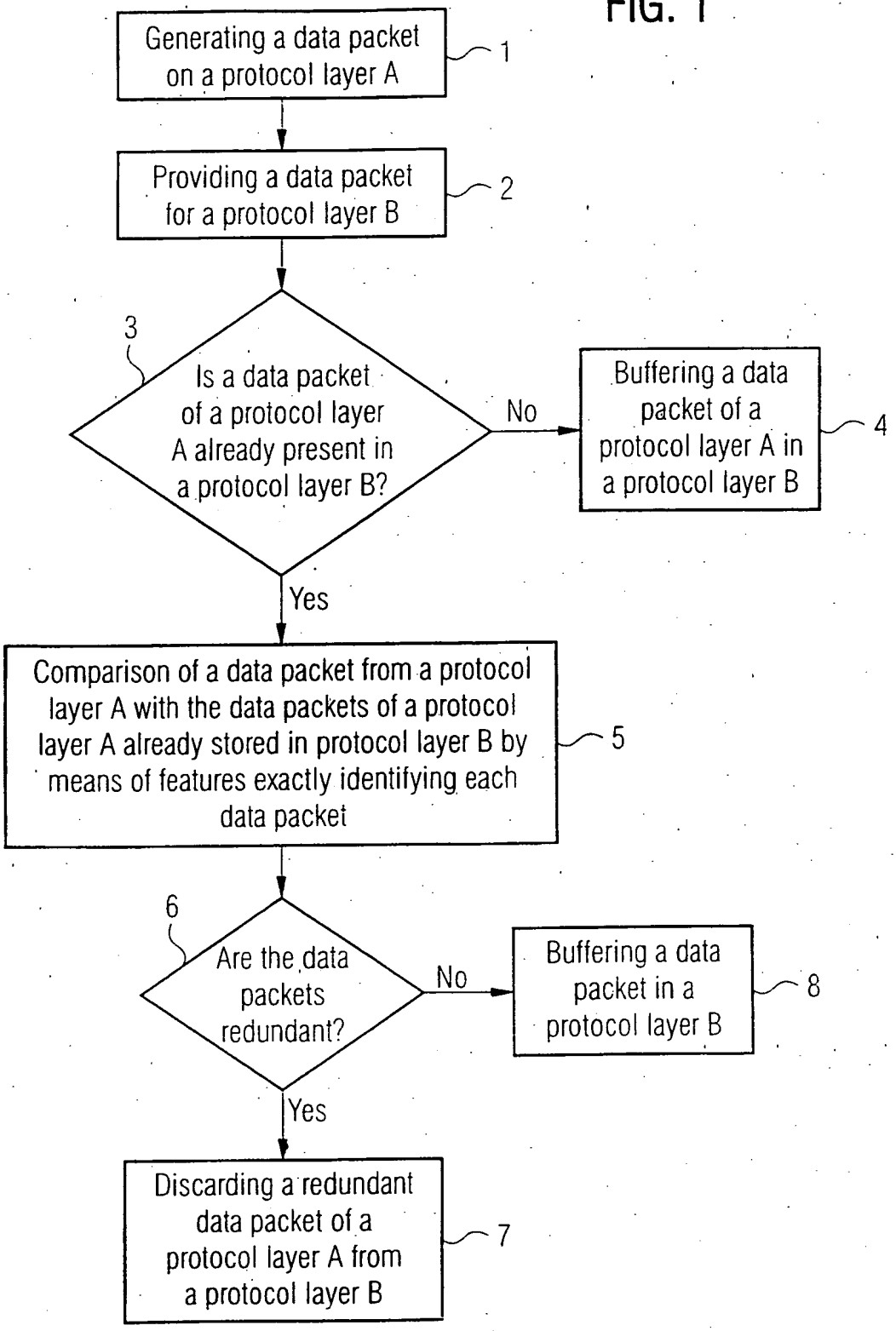


FIG. 2

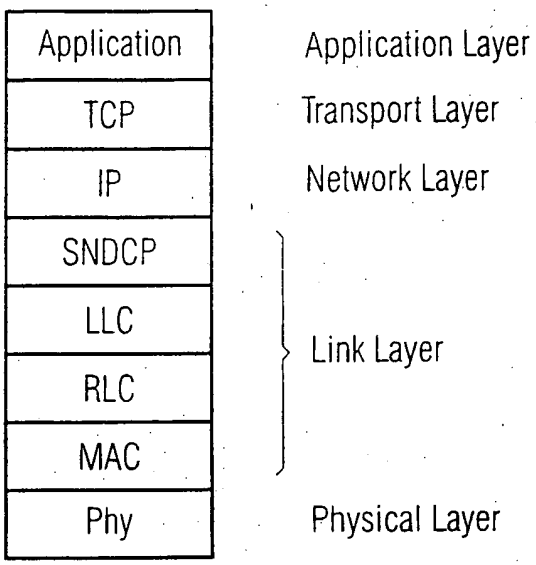
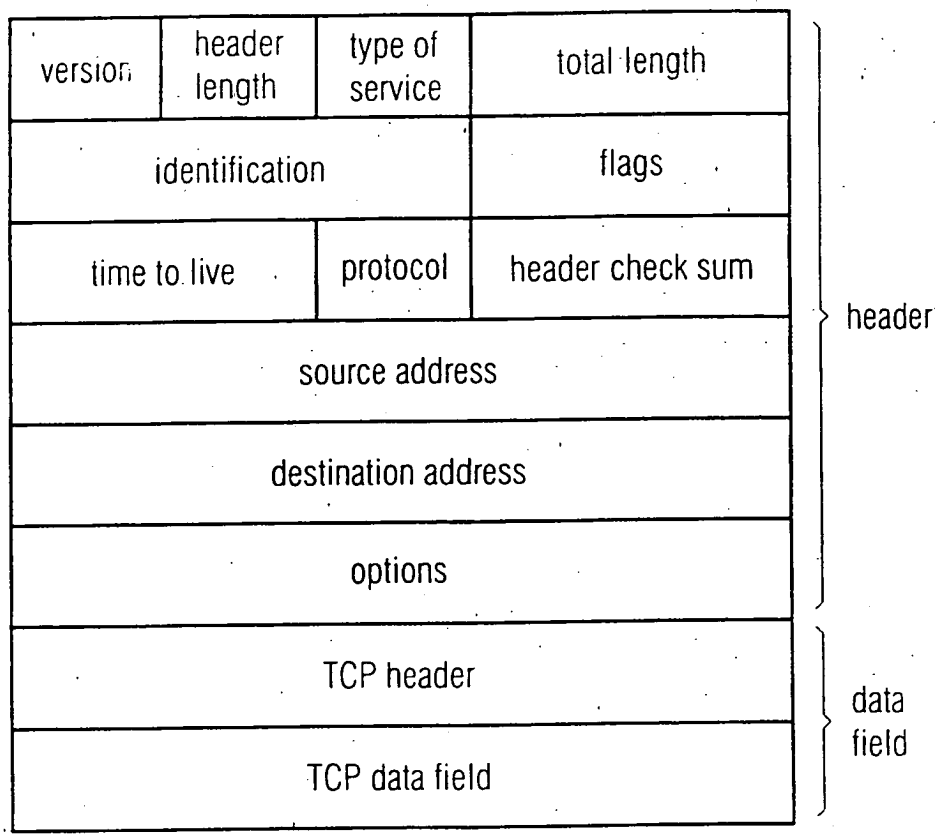


FIG. 3

source port number	destination port number	
sequence number		
acknowledgment		
header length	flags	window size
TCP check sum		urgent pointer

FIG. 4


version
header length
type of service
total length
identification
flags
time to live
protocol
header check sum
source address
destination address
options
TCP header
TCP data field
header
data field

INTERNATIONAL SEARCH REPORT

Int· ional Application No
PCT/EP 00/10693

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/56

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DESIMONE A ET AL: "THROUGHPUT PERFORMANCE OF TRANSPORT-LAYER PROTOCOLS OVER WIRELESS LANS" PROCEEDINGS OF THE GLOBAL TELECOMMUNICATIONS CONFERENCE (GLOBECOM),US,NEW YORK, IEEE, vol. -, 1993, pages 542-549, XP000428113 page 1, left-hand column, line 1 - line 21 page 1, right-hand column, line 37 -page 2, left-hand column, line 41 page 2, right-hand column, line 5 - line 18 page 6, left-hand column, line 8 - line 17; figure 6 page 7, right-hand column, line 11 -page 8, left-hand column, line 18 --- -/--	1,17,20

[X] Further documents are listed in the continuation of box C. [X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 February 2001	07/03/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Brichau, G

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/EP 00/10693

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CHUN B -G ET AL: "AUXILIARY TIMEOUT AND SELECTIVE PACKET DISCARD SCHEMES TO IMPROVE TCP PERFORMANCE IN PCN ENVIRONMENT" IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC),US,NEW YORK, NY: IEEE. 1997. pages 381-385. XP000740265 ISBN: 0-7803-3926-6 page 381, left-hand column. line 1 - line 27 page 383, left-hand column. line 32 -right-hand column, line 32 ---	1,17,20
A	WO 99 37071 A (QUALCOMM INC) 22 July 1999 (1999-07-22) page 1, line 27 -page 2. line 36 page 5, line 26 -page 6. line 28 page 13, line 18 -page 14, line 3 -----	1,17,20

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int ional Application No
PCT/EP 00/10693

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9937071 A	22-07-1999	US 6118765 A	12-09-2000
		AU 2318399 A	02-08-1999

Form PCT/ISA/210 (patent family annex) (July 1992)

BNSDOCID: <WO ____ 0137493A1 I >